

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-270017

(43)Date of publication of application : 14.10.1997

-----  
(51)Int.Cl. G06T 11/00  
G09G 5/36

-----  
(21)Application number : 09-019355 (71)Applicant : TOSHIBA CORP

(22)Date of filing : 31.01.1997 (72)Inventor : KUNIMATSU ATSUSHI

-----  
(30)Priority

Priority number : 08 17896

Priority date : 02.02.1996

Priority country : JP

-----  
(54) COORDINATE CONVERSION PROCESSING CIRCUIT THAT PERFORMS  
COORDINATE CONVERSION AND CLIPPING JUDGEMENT

(57)Abstract:

PROBLEM TO BE SOLVED: To efficiently decide clipping after coordinate conversion by providing a specific number of comparing circuits which judges whether or not coordinate are in a display area by comparing coordinate values of an output coordinate vector with the value of the output coordinate vector for a specific number of coordinate

converting circuits.

SOLUTION: A floating-point processor has source buses 1 and 2 and a destination bus 3 and a coordinate conversion processing circuit has three computing element 5x, 5y, and 5z containing comparators 4x, 4y, and 4z for clipping, a computing element 5w containing no comparator, and a clipping register 6. The computing element 5x, 5y, and 5w compares the coordinate values of the output coordinate vectors with the values of the output coordinate vectors to judge whether or not the coordinates are in the display area. Consequently, instruction for performing clipping judgement can be decreased in number to shorten the time needed for the clipping judgement processing.

-----  
LEGAL STATUS [Date of request for examination] 11.12.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3655723

[Date of registration] 11.03.2005

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

\* NOTICES \*

JPO and INPIT are not responsible for any  
damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not  
reflect the original precisely.

2.\*\*\*\* shows the word which can not be translated.

3.In the drawings, any words are not translated.

---

## CLAIMS

---

[Claim(s)]

[Claim 1] Four coordinate transformation circuits which perform transparent transformation of an input coordinate vector  $(x, y, z_1)$ , and output each element of an output coordinate vector  $(x', y', z', w')$  as the conversion result, By being prepared about each of said coordinate transformation circuit, and comparing each coordinate value of said output coordinate vector  $(x', y', z')$  with the value of said output coordinate vector  $(w')$  The coordinate transformation processing circuit equipped with the clipping register which stores the output from the no settled in the viewing area about each coordinate, three comparator circuits to judge, and said comparator circuit.

[Claim 2] The value of said output coordinate vector is expressed in the field which shows a sign and an absolute value, respectively. Said comparator circuit The comparator which compares the fields which show an absolute value, and the output from said comparator and the 1st circuit which outputs the AND of each coordinate value of said output coordinate vector ( $x'$ ,  $y'$ ,  $z'$ ), The coordinate transformation processing circuit according to claim 1 characterized by consisting of a reversal signal of the output from said comparator, and the 2nd circuit which outputs the AND of each coordinate value of said output coordinate vector ( $x'$ ,  $y'$ ,  $z'$ ).

[Claim 3] Said clipping register is a coordinate transformation processing circuit according to claim 1 which is a shift register and is evacuating-by shift action characterized by the output from said comparator circuit.

[Claim 4] In  $x' > |w'|$  or  $x' < -|w'|$ , said comparator circuit outputs the signal which shows that  $x'$  is outside a viewing area. The coordinate transformation processing circuit according to claim 1 by which it is outputting [ in  $y' > |w'|$  or  $y' < -|w'|$ , output the signal which shows that  $y'$  is outside a viewing area, and / in  $z' > |w'|$  or  $z' < -|w'|$  ]-signal which shows that  $z'$  is outside viewing area characterized.

[Claim 5] Each of said comparator circuit is a coordinate transformation processing circuit according to claim 4 characterized by outputting the 2-bit recognition signal which shows any of the case where each coordinate value of said output coordinate vector ( $x'$ ,  $y'$ ,  $z'$ ) is settled in the viewing area about each coordinate, the case where it has separated in the positive direction of the viewing area, and the case where it has separated in the negative direction of the viewing area they are.

[Claim 6] Said clipping register is a coordinate transformation processing circuit according to claim 5 by which is the shift register of  $6 \times n$  bit ( $n > 1$ ), and it is evacuating [ the amount of / the amount of / for 6 bits / register section receives the recognition signal of said comparator circuit, and / one by one / to other parts ]-by 6-bit shift action characterized.

[Claim 7] Said clipping register is a coordinate transformation processing circuit according to claim 6 by which is at least 18-bit shift register, and it is connecting [ the output circuit which outputs this 18-bit AND ] characterized.

---

## DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the coordinate transformation processing circuit of the floating-point-processor unit which performs coordinate transformation used for graphics drawing processing etc.

[0002]

[Description of the Prior Art] In graphics drawing processing, the top-most-vertices coordinate of a polygon is changed into the two-dimensional coordinate for displays from a three-dimension coordinate. This coordinate transformation is called for by calculating the product of the vector of a matrix (4x4) and four elements. The two-dimensional coordinate after coordinate transformation needs to perform the clipping judging which confirms whether enter within the limits of the viewing area.

[0003] In the above-mentioned graphics drawing processing,

入力座標ベクトル : (x, y, z, 1)  
 出力座標ベクトル : (x', y', z', w')  
 座標変換・透視変換行列 : (a, b, c, d)  
                                   (e, f, g, h)  
                                   (i, j, k, l)  
                                   (m, n, o, p)

It is [Equation 1] when it carries out.

$$(x', y', z', w') = (x, y, z, 1) (a, b, c, d)$$

$$(e, f, g, h)$$

$$(i, j, k, l)$$

$$(m, n, o, p)$$

$$=(ax+ey+iz+lm, bx+fy+jz+ln, cx+gy+kz+lo, dx+hy+lz+lp)$$

Since it becomes, if four sum-of-products computing elements are repeated 4 times to juxtaposition, coordinate transformation can be performed at a high speed. After this coordinate transformation finishes, the clipping check of whether the changed coordinate is within the limits of a viewing area is performed.

[0004] It is [Equation 2] when one after coordinate transformation ( $x'$ ,  $y'$ ,  $z'$ ,  $w'$ ) is considered in fact.

- $|w'| < x' < |w'|$  -- (one formula)
- $|w'| < y' < |w'|$  -- (two formulas)
- $|w'| < z' < |w'|$  -- (three formulas)

When all upper types are materialized, it becomes in a field. This will be called the clipping judging of a point.

[0005] In outside a field, it can be confirmed in which direction it has come out on condition that the following.

[0006]

[Equation 3]

| Outside of the field of the  $x' < -|w'|$ :  $x$  negative direction -- (four formulas)

| Outside of the field of the  $x' > |w'|$ :  $x$  forward direction -- (five formulas)



| Outside of the field of the  $y' < -|w'|:y$  negative direction -- (six formulas)

| Outside of the field of the  $y' > |w'|:y$  forward direction -- (seven formulas)

| Outside of the field of the  $z' < -|w'|:z$  negative direction -- (eight formulas)

| Outside of the field of the  $z' > |w'|:z$  forward direction -- (nine formulas)

In drawing a polygon (n square shape), it performs the point clipping judging for n top-most vertices. Consequently, when it is in a field altogether, it can judge with the polygon being in a field, and displays as it is. At least one will clip and display the polygon, if top-most vertices are outside a field. Moreover, when all the top-most vertices of the polygon are outside a field, there are two kinds of cases. The polygon will not be displayed, if all top-most vertices fill (4 Formula) if all top-most vertices are outside the field of the fixed direction for example. Other than this, top-most vertices 1 fill (4 Formula), and except it, when filling (five formulas), the polygon is clipped and displayed.

[0007] An example of the floating point processor which performs such coordinate transformation and a clipping judging is shown in drawing 5.

[0008] The floating point processor shown in drawing 5 is designed as a co-processor, and the fetch of an instruction, decoding, etc. are performed by the main processor side. A TRUE/FALSE signal is sent to a main processor side

from a co-processor side, and, as for conditional branching, conditional branching is performed by the signal by the main processor side.

[0009] This floating point processor has the 128-bit destination bus 3 as well as the 128-bit source bus 1 and the source bus 2, and has the computing elements 7x, 7y, 7z, and 7w as shown in the register 4 as shown in drawing 6 , and drawing 7 . 32 bits of each 128-bit bus are divided into each four groups. This will be called x lines, y lines, z lines, and w lines.

[0010] By the source bus 2 and destination bus 3 x buses are connected to the register (R0, R4, R8, --) which the register number of a register 4 can divide among 4. y buses are connected to the register (R1, R5, R9, --) which will remain one if a register number divides by 4. z buses will be connected to the register (R2, R6, R10, --) which remains two if a register number divides by 4, and w buses are connected to the register (R3, R7, R11, --) which remains three if a register number divides by 4. All buses are connected to all registers and the source bus 1 has crossbar composition.

[0011] Computing elements 7x, 7y, 7z, and 7w can constitute a sum-of-products computing element (FMACx, FMACy, FMACz, FMACw), respectively, and can perform a sum-of-products operation, multiplication, addition, and subtraction by

4 juxtaposition. Computing-element 7w performs special operations, such as a comparison operation.

[0012] By connecting, as shown in drawing 5 , the matrix in coordinate transformation (4x4) is calculable at a high speed. The instruction and example program which are used for below by matrix count are shown.

[0013] Description: R numeric value shows the register of a register number numeric value.

[0014] The multiplication of four FMUL instructions is performed to juxtaposition.

```
FMUL  R 2 0 - R 2 3, R 1 6, R 0 - R 3
      R 2 0 ← R 1 6 × R 0
      R 2 1 ← R 1 6 × R 1
      R 2 2 ← R 1 6 × R 2
      R 2 3 ← R 1 6 × R 3
```

[0015] Example :

The multiplication of four FMAC instructions is performed to juxtaposition.

```
FMUL  R 2 0 - R 2 3, R 1 6, R 0 - R 3
      R 2 0 ← R 2 0 + R 1 6 × R 0
      R 2 1 ← R 2 1 + R 1 6 × R 1
      R 2 2 ← R 2 2 + R 1 6 × R 2
      R 2 3 ← R 2 3 + R 1 6 × R 3
```

[0016] Example :

The sign of a FNEG instruction floating point number is reversed.

```
FNEG  R 0, R 1
      R 0 = - (R 1)
```

[0017] Example :

A FCMP instruction floating point number is compared. A comparison result is sent to a processor side using a TRUE/FALSE signal.

[0018] Example: FCMP eq A TRUE signal will be sent if R0, R1R0, and R1 are equal. Otherwise, a FALSE signal is sent.

[0019] FCMP ne A TRUE signal will be sent if R0, R1R0, and R1 become equal. Otherwise, a FALSE signal is sent.

[0020] FCMP gt A TRUE signal will be sent if R0 and R1R0 are larger than R1. Otherwise, a FALSE signal is sent.

[0021] FCMP le A TRUE signal will be sent if R0 and R1R0 become one or less [ R ]. Otherwise, a FALSE signal is sent.

[0022] FCMP lt A TRUE signal will be sent if R0 and R1R0 are smaller than R1. Otherwise, a FALSE signal is sent.

[0023] FCMP eq A TRUE signal will be sent if R0 and R1R0 become one or more [ R ]. Otherwise, a FALSE signal is sent.

[0024] If a TRUE signal is sent from a BCT instruction co-processor, it will branch to LABEL.

[0025] example: -- BCT label -- it is as follows when a triangle describes the

program which performs the coordinate transformation of a polygon, and a clipping judging using these instructions.

[0026] - top-most-vertices 1: (R0, R1, R2, R3) of the triangle polygon before correspondence of each data and a register - conversion

Top-most vertices 2 of the triangle polygon before conversion : (R4, R5, R6, R7)

Top-most vertices 3 of the triangle polygon before conversion : (R8, R9, R10, R11)

透視変換行列 :

R 1 6, R 1 7, R 1 8, R 1 9  
2 0, R 2 1, R 2 2, R 2 3  
2 4, R 2 5, R 2 6, R 2 7  
2 8, R 2 9, R 3 0, R 3 1

透視変換行列 :

$$\begin{pmatrix} R 1 6, R 1 7, R 1 8, R 1 9 \\ R 2 0, R 2 1, R 2 2, R 2 3 \\ R 2 4, R 2 5, R 2 6, R 2 7 \\ R 2 8, R 2 9, R 3 0, R 3 1 \end{pmatrix}$$

Top-most vertices 1 of the triangle polygon after conversion : (R32, R33, R34, R35)

Top-most vertices 2 of the triangle polygon after conversion : (R36, R37, R38, R39)

Top-most vertices 3 of the triangle polygon after conversion : (R40, R41, R42, R43)

..... From here to a program listing .. list 1 ..... ; The coordinate transformation FMUL of top-most vertices 1 R32-R35, R0, and R16-R19FMAC R32-R35, R1, and R20-R23FMAC R32-R35, R2, and R24-R27FMAC R32-R35, R3, and R28-R31 ; coordinate transformation FMUL of top-most vertices 2 R36-R39, R4, R16-R19FMAC R36-R39, R5, R20-R23FMAC R36-R39, R6, R24-R27FMAC R36-R39, R7, and R28-R31 ; coordinate transformation FMUL of top-most vertices 3 R40-R43, R8, R16-R19FMAC R40-R43, R9, R20-R23FMAC R40-R43, R10, R24-R27FMAC R40-R43, R11, and R28-R31 ; clipping judging (w assumes that it is forward)

; Top-most-vertices 1FCMP gt R32, R35 ; It is if it is top-most-vertices 1:x>w. -- BCT label ; It is Jump FCMP to clipping processing. gt R33, R35 ; It is if it is top-most-vertices 1:y>w. -- BCT label ; It is Jump FCMP to clipping processing. gt R34, R35 ; It is if it is top-most-vertices 1:z>w. -- BCT label ; It is Jump FNEG to clipping processing. gt R35, R35 ; R35= - (R35)

FCMP lt R32, R35 ; It is if it is top-most-vertices 1:x<-w. -- BCT label ; It is Jump FCMP to clipping processing. lt R33, R35 ; It is if it is top-most-vertices 1:y<-w. -- BCT label ; It is Jump FCMP to clipping processing. lt R34, R35 ; It is if it is top-most-vertices 1:z<-w. -- BCT label ; Jump to clipping processing. ;

Top-most-vertices 2FCMP gt R36, R39 ; It is if it is top-most-vertices 2:x>w. --  
BCT label ; It is Jump FCMP to clipping processing. gt R37, R39 ; It is if it is  
top-most-vertices 2:y>w. -- BCT label ; It is Jump FCMP to clipping processing.  
gt R38, R39 ; It is if it is top-most-vertices 2:z>w. -- BCT label ; It is Jump FNEG  
to clipping processing. R39, R39 ; R39= - (R39)

FCMP It R36, R39 ; It is if it is top-most-vertices 2:x<-w. -- BCT label ; It is Jump  
FCMP to clipping processing. It R37, R39 ; It is if it is top-most-vertices 2:y<-w. --  
BCT label ; It is Jump FCMP to clipping processing. It R38, R39 ; It is if it is  
top-most-vertices 2:z<-w. -- BCT label ; Jump to clipping processing. ;

Top-most-vertices 3FCMP gt R40, R43 ; It is if it is top-most-vertices 3:x>w. --  
BCT label ; It is Jump FCMP to clipping processing. gt R41, R43 ; It is if it is  
top-most-vertices 3:y>w. -- BCT label ; It is Jump FCMP to clipping processing.  
gt R42, R43 ; It is if it is top-most-vertices 3:z>w. -- BCT label ; It is Jump FNEG  
to clipping processing. R43, R43 ; R43= - (R43)

FCMP It R40, R43 ; It is if it is top-most-vertices 3:x<-w. -- BCT label ; It is Jump  
FCMP to clipping processing. It R41, R43 ; It is if it is top-most-vertices 3:y<-w. --  
BCT label ; It is Jump FCMP to clipping processing. It R42, R43 ; It is if it is  
top-most-vertices 3:z<-w. -- BCT label ; It jumps [ 1 ... [0027] ] to clipping

processing..... It is a program listing so far.. list

[Problem(s) to be Solved by the Invention] As explained above, in order to combine the compare instruction of two numeric values, and a conditional-branching instruction in the conventional processor which performs coordinate transformation in graphics drawing processing, many instructions were needed for the comparison for a clipping judging, and conditional branching. For this reason, the fault that the coordinate transformation engine performance accompanied by a clipping judging will fall was caused.

[0028] Then, this invention is made in view of the above, and the place made into that purpose is to offer the coordinate transformation processing circuit which performs efficiently the clipping judging after the coordinate transformation in graphics drawing processing.

[0029]

[Means for Solving the Problem] In order to attain the above-mentioned purpose, invention according to claim 1 Four coordinate transformation circuits which perform transparent transformation of an input coordinate vector  $(x, y, z_1)$ , and output each element of the output coordinate vector  $(x', y', z', w')$  as the conversion result, By being prepared about each of said coordinate



transformation circuit, and comparing each coordinate value of said output coordinate vector  $(x', y', z')$  with the value of said output coordinate vector  $(w')$  It has the clipping register which stores the output from the no settled in the viewing area, three comparator circuits to judge, and said comparator circuit, and is constituted by each coordinate.

[0030] Invention according to claim 2 is expressed further in the field where the value of said output coordinate vector shows a sign and an absolute value, respectively. Said comparator circuit The comparator which compares the fields which show an absolute value, and the output from said comparator and the 1st circuit which outputs the AND of one coordinate value of said output coordinate vector  $(x', y', z')$ , It consists of a reversal signal of the output from said comparator, and the 2nd circuit which outputs the AND of one coordinate value of said output coordinate vector  $(x', y', z')$ .

[0031] Said clipping register is a shift register and invention according to claim 3 is characterized by evacuating the output from said comparator circuit by the shift action.

[0032] As for said comparator circuit, invention according to claim 4 outputs the signal with which  $x'$  shows that it is outside a viewing area in  $x' > |w'|$  or  $x' < -|w'|$ . It

considers as the outputting [ in  $y' > |w'|$  or  $y' < -|w'|$ , output the signal which shows that  $y'$  is outside a viewing area, and / in  $z' > |w'|$  or  $z' < -|w'|$  ]-signal which shows that  $z'$  is outside viewing area description.

[0033] Invention according to claim 5 is characterized by each of said comparator circuit outputting the 2-bit recognition signal which shows any of the case where each coordinate value of said output coordinate vector ( $x'$ ,  $y'$ ,  $z'$ ) is settled in the viewing area about each coordinate, the case where it has separated in the positive direction of the viewing area, and the case where it has separated in the negative direction of the viewing area they are.

[0034] Said clipping register is a shift register of  $6 \times n$  bit ( $n > 1$ ), the amount of [ for 6 bits ] register section receives the recognition signal of said comparator circuit, and invention according to claim 6 is evacuating [ one by one / to other parts ]-by 6-bit shift action characterized by it.

[0035] Said clipping register is at least 18-bit shift register, and invention according to claim 7 is connecting [ the output circuit which outputs this 18-bit AND ] characterized by it.

[0036]

[Embodiment of the Invention] Hereafter, the gestalt of implementation of this

invention is explained using a drawing.

[0037] The graphics operation system using the coordinate transformation processing circuit by one example of this invention is shown in drawing 1 .

[0038] In this system, main memory 5 is connected with the processor 1 and the graphics processor unit (it is hereafter described as "GPU".) 3 in Main Bath 7, respectively, and the graphics memory 9 is further connected to GPU3 by graphics bus 11.

[0039] GPU3 has the pretreatment section 13 and the Main processing section 15, the Main processing section 15 uses DDA, and carries out the rendering of the pixel of an output image using the texture information read in the graphics memory 9, and writes a result in the frame buffer on a graphics memory 9. moreover, the pretreatment section 13 -- the initial value of DDA of the Main processing, and difference -- parameters, such as a value, are calculated.

[0040] A processor 1 reads a program from main memory 5 with the processor core 17, and performs. The processor core 17 makes a two-dimensional coordinate, color information, etc. of top-most vertices on a polygon which should draw on a screen according to a program (it is hereafter described as "GPU command information".). A two-dimensional coordinate is made from a

three-dimension coordinate using the coordinate transformation unit 19 in that case. Furthermore, a processor 1 adds additional information (discernment command of what kind of polygon to draw) to the above-mentioned GPU command information according to a program, and delivery and GPU3 perform drawing to GPU3 through Main Bus 7 as a GPU command. In addition, also when sending information using the bus of dedication, without using Main Bus 7, it may think. Moreover, when the speed which makes the GPU command of a processor 1, and the drawing speed of GPU3 do not balance, on main memory 5, the GPU command is buffered (either is late) and it absorbs the speed difference.

[ it ] [ it ]

[0041] The configuration of the coordinate transformation processing circuit 19 is shown in drawing 2 . Compared with the coordinate transformation processing circuit shown in drawing 5 , the coordinate transformation processing circuit of this operation gestalt is equipped with the clipping register 6 indicated to be the computing elements 5x, 5y, and 5z which contained the comparators 4x, 4y, and 4z for clipping which are shown in drawing 3 in addition to a function equivalent to the computing elements 7x, 7y, and 7z shown in drawing 5 to drawing 4 , and is constituted, and other configurations are the same as that of drawing 5 .

[0042] In drawing 3 the clipping comparators 4x, 4y, and 4z. The output coordinate vector  $(x_n, y_n, z_n, \text{ and } w_n)$  by the floating point of the top-most vertices  $n$  after coordinate transformation ( $2 \leq n \leq 3$ ) is received.  $x_n$   $w_n$  Compare and  $x_n > |w_n|$ ,  $x_n < -|w_n|$ ,  $x_n \geq |w_n|$ , or  $x_n \leq -|w_n|$  is distinguished.  $y_n$   $w_n$  Compare and  $y_n > |w_n|$ ,  $y_n < -|w_n|$ ,  $y_n \geq |w_n|$ , or  $y_n \leq -|w_n|$  is distinguished.  $z_n$   $w_n$  Compare and  $z_n > |w_n|$ ,  $z_n < -|w_n|$ ,  $z_n \geq |w_n|$ , or  $z_n \leq -|w_n|$  is distinguished.

The input registers 51 and 52 which perform actuation which outputs each distinction result by one compare instruction, and hold the output coordinate vector  $(x_n, y_n, z_n, \text{ and } w_n)$  of the top-most vertices  $n$  after coordinate transformation including a sign. The output coordinate vector  $(x_n, y_n, \text{ and } z_n)$  held at the input register 51, and output coordinate vector  $w_n$  held at the input register 52. The comparator 53 which compares size relation, respectively, The reversal value of the sign of the output coordinate vector held at the comparison result and input register 51 of a comparator 53 is inputted, and it is  $x_n > w_n$ . Or  $y_n > w_n$  Or  $z_n > w_n$  Or  $x_n \geq w_n$  Or  $y_n \geq w_n$  Or  $z_n \geq w_n$ . The AND (AND) gate 54 to distinguish, The sign of the output coordinate vector held at the comparison result and input register 51 of a comparator 53 is inputted.  $x_n < -w_n$  or  $y_n < -w_n$  Or  $z_n < -w_n$  Or  $x_n \leq -w_n$  Or  $y_n \leq -w_n$  Or  $z_n \leq -w_n$ . It has the AND gate 55 to

distinguish and is constituted.

[0043] The clipping register 6 is equipped with a shift register 61 and the OR (OR) gate 62 in drawing 4 . The OR (OR) gate 62 outputs the 18-bit bit product on the right-hand side of a shift register 61.

[0044] By activation of said compare instruction, a shift register 61 performs a left shift by 6 bits, in order to receive each distinction result outputted from the clipping comparator of computing elements  $5x$ ,  $5y$ , and  $5z$ . that is, the inside of a series of processings -- the above of top-most vertices ( $n-1$ ) -- after shifting the distinction result of a force coordinate vector ( $x_{n-1}$ ,  $y_{n-1}$ ,  $z_{n-1}$ , and  $w_{n-1}$ ), the following clipping distinction result is received to 6 bits of the vacant right-hand side. And when the top-most vertices which the clipping judging of whether the top-most vertices where distinction by the clipping comparator is materialized exist is performed with a shift register 61, and should carry out clipping from the distinction result of the top-most vertices of  $n$  square shape (triangle) polygon by which shift in of the distinction result was carried out, and it was held in the phase which gathered by  $n$  top-most vertices (here three top-most vertices) exist, the signal which performs the branch instruction which performs clipping processing outputs.

[0045] In the clipping comparator shown in drawing 3 , if the input of an input register 51 is computing-element  $5x$ ,  $x$  values are computing-element  $5y$  and  $y$  value is computing-element  $5z$ ,  $z$  value is inputted.  $w$  value with the input of an input register 52 same at every computing element  $5x$ ,  $5y$ , and  $5z$  is inputted.

[0046] If it explains on behalf of Input  $x$ , the sign bit of Input  $x$  and the sign bit of Input  $w$  will be separated, and only a numeric field will be inputted into a comparator 53. Here, the numeric field of Input  $x$ , and  $y$  and  $z$  is considered as an input 1, and the numeric field of Input  $w$  is considered as an input 2. The numeric field consists of the exponent part and the fixed point part of the floating point here.

[0047] In the input  $1 > \text{input } 2$ , the comparison result of a comparator 53 becomes with "0", when that is not right, "1" and. The input  $1 \geq \text{input } 2$  is sufficient as comparative conditions. Furthermore, Input  $x$  and the reversal value of the sign bit of  $y$  and  $z$  are considered as one input of the AND gate 54, and the sign bit of Input  $x$ , and  $y$  and  $z$  is considered as one input of the AND gate 55.

[0048] In the configuration shown in drawing 4 , a comparator 53 compares the absolute value of Input  $x$  and Input  $w$ . | +Clipping which is the output of the AND gate 54 in  $x > |w|$  at the time of  $x \geq 0$  (that is,  $x > w$ ) Info. is set to "1." | -Clipping

which is the output of the AND gate 55 in  $x > |w|$  at the time of  $x < 0$  (that is,  $x < -w$ ) Info. is set to "1" (it is assumed that  $w$  is forward).

[0049] Next, the output from each comparator for clipping of computing elements  $5x$ ,  $5y$ ,  $5z$ , and  $5w$ ,  $+x$  Clipping Info.,  $+y$  Clipping Info.,  $+z$  Clipping Info.,  $-x$  Clipping Info.,  $-y$  Clipping Info.,  $-z$  Clipping Info. is considered as the input to the clipping register 6.

[0050] As for the comparator for clipping, the comparator for clipping of computing elements  $5x$ ,  $5y$ , and  $5z$  is performed by coincidence with a FCLIP instruction. A FCLIP instruction performs a clipping comparison after a 6-bit left shift, and a clipping judging is performed. It is as [ detailed ] follows.

[0051] Example FCLIP of FCLIP instruction actuation R0-R3 and R4(1) He is a 6-bit left shift about the first actuation shift register 61.

[0052] (2) The clipping comparator of computing-element  $5x$  compares the second actuation R0 (input  $x$ ) and R4 (input  $w$ ). The clipping comparator of computing-element  $5y$  compares R1 (input  $y$ ) and R4 (input  $w$ ). The clipping comparator of computing-element  $5z$  compares R2 (input  $z$ ) and R4 (input  $w$ ).

[0053] (3) If there is one "at least 1" in 18 bits of low order of the third actuation clipping judging shift register 61, a TRUE signal will be sent to a main processor.



[0054] In addition, since the first actuation of the above and the second actuation are independent, they are parallel, and they are performed.

[0055] As the shift register 61 which constitutes the clipping register 6 is shown in drawing 5 , it is a 32-bit register and 6 bits of least significant are connected to the output of a clipping comparator. The 0th bit is +Clipping. To Info., the 1st bit is +Clipping. To Info., the 2nd bit is +Clipping. To Info., the 3rd bit is +Clipping. To Info., the 4th bit is +Clipping. To Info., the 5th bit is +Clipping. It connects with Info., respectively.

[0056] Since it acts as the 6-bit left shift of the shift register 61 whenever a FCLIP instruction is executed To the 0-5th bit, the clipping information on the top-most vertices in front of one always to the 6-11th bit [ the clipping information on the newest top-most vertices ] The clipping information on the top-most vertices in front of three will be contained into the 18-23rd bit, and the clipping information on the top-most vertices in front of two will be [ the clipping information on the top-most vertices in front of four ] contained in the 12-17th bit at the 24-29th bit.

[0057] That is, branch instruction is executed so that in the case of a triangle polygon a TRUE signal may be outputted to a main processor and it may perform clipping processing for the information for three top-most vertices, if it

judges whether there is one "at least 1" among 0th bit - 17 bits using the OR gate 62 and there is "1" since the clipping judging of a polygon is performed.

[0058] By adopting such a configuration, by the comparator for clipping, the compare instruction for a clipping judging which two needed is made to one, and it can process at a high speed. Moreover, with the clipping register 6, the clipping information for newest 5 top-most vertices can always be held, and the clipping information for five top-most vertices of a pentagon polygon can be held from the clipping information for three top-most vertices of a triangle polygon. Furthermore, list mentioned above by using the comparator for clipping and clipping register which were mentioned above 1 is list shown below. It becomes like 2 and the number of steps of the program of a clipping judging can be shortened.

[0059]

..... From here to a program listing .. list 2 ..... ; The coordinate transformation FMUL of top-most vertices 1 R32-R35, R0, and R16-R19FMAC R32-R35, R1, and R20-R23FMAC R32-R35, R2, and R24-R27FMAC R32-R35, R3, and R28-R31 ; coordinate transformation FMUL of top-most vertices 2 R36-R39, R4, R16-R19FMAC R36-R39, R5, R20-R23FMAC R36-R39, R6, R24-R27FMAC

R36-R39, R7, and R28-R31 ; coordinate transformation FMUL of top-most vertices 3 R40-R43, R8, R16-R19FMAC R40-R43, R9, R20-R23FMAC R40-R43, R10, R24-R27FMAC R40-R43, R11, and R28-R31 ; clipping judging (w assumes that it is forward)

FCLIP R32-R34, R35; it is FCLIP to a clipping register about the clipping information on top-most vertices 1. R36-R38, R39; it is FCLIP to a clipping register about the clipping information on top-most vertices 2. R40-R32, R43; the clipping information on top-most vertices 2 It is BCT to a clipping register. label ; Jump to clipping processing based on the information for newest 3 top-most vertices..... It is a program listing so far.. list 2 ... Again in the clipping register 6 of this example Since the clipping information on each top-most vertices (+x, -x, +y, -y, +z, -z) is held, it can be confirmed efficiently whether all applicable polygons are outside a clipping region.

[0060] For example, in a triangle polygon, when +x clipping information bits of top-most vertices 1, +x clipping information bits of top-most vertices 2, and all the +x clipping information bits of top-most vertices 3 are "1", it can be judged that it is not necessary to be and indicate the whole of the polygon outside a clipping region. In this case, what is necessary is just to transpose the OR gate

62 to the logic gate which takes the AND of  $+x$  of top-most vertices 1,  $+x$  of top-most vertices 2, and  $+x$  of top-most vertices 3.

[0061] In addition, although the top-most vertices  $n$  after coordinate transformation were made into three pieces ( $2 \leq n \leq 3$ ), it is obvious that the polygon of a polygon with the top-most vertices ( $n \geq 3$ ) of the number of arbitration can completely be dealt with similarly here. That is, what is necessary is to increase the input number of bits of the OR gate 62, and just to enlarge bit width of face of a shift register 61 if needed.

[0062]

[Effect of the Invention] According to [ as explained above ] this invention, it is the output coordinate vector  $(x_n, y_n, \text{ and } z_n)$  and  $w_n$  after coordinate transformation. Since the instruction for a clipping judging can be lessened since the configuration which performs each comparison with one instruction was adopted, and the number of steps of the program of a clipping judging can be reduced, the time amount concerning clipping judging processing can be shortened. Moreover, since the configuration which brings together the comparison information for a clipping judging in a specified register, and performs a clipping judging was adopted, the clipping judging for one polygon

can be performed efficiently.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1] The graphics operation system using the coordinate transformation

processing circuit by one example of this invention is shown.

[Drawing 2] It is drawing showing the configuration of the coordinate transformation processing circuit concerning 1 operation gestalt of invention.

[Drawing 3] It is drawing showing the configuration of a clipping comparator.

[Drawing 4] It is drawing showing the configuration of a clipping register.

[Drawing 5] It is drawing showing the configuration of the conventional coordinate transformation processing circuit which performs coordinate transformation in graphics operation.

[Drawing 6] It is drawing showing the configuration of the register shown in drawing 5 .

[Drawing 7] It is drawing showing the configuration of the computing element shown in drawing 5 .

[Description of Notations]

1 Two Source bus

3 Destination Bus

4 Register

5x, 5y, 5z, 5w, 7x, 7y, 7z, 7w Computing element

6 Clipping Register

51 52 Input register

53 Comparator

54, 55, 62 Logic gate

61 Shift Register

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-270017

(43)公開日 平成9年(1997)10月14日

(51) Int. Cl. <sup>8</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 T 11/00			G 0 6 F 15/72	3 8 0
G 0 9 G 5/36	5 2 0		G 0 9 G 5/36	5 2 0 P

審査請求 未請求 請求項の数7 OL (全 12 頁)

(21)出願番号 特願平9-19355

(22)出願日 平成9年(1997)1月31日

(31)優先權主張番号 特願平8-17896

(32)優先日 平8(1996)2月2日

(33)優先権主張国 日本 (J P)

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 国松 敦

神奈川県川崎市幸区堀川町580番1号 株

式会社東芝半導体システム技術センター内

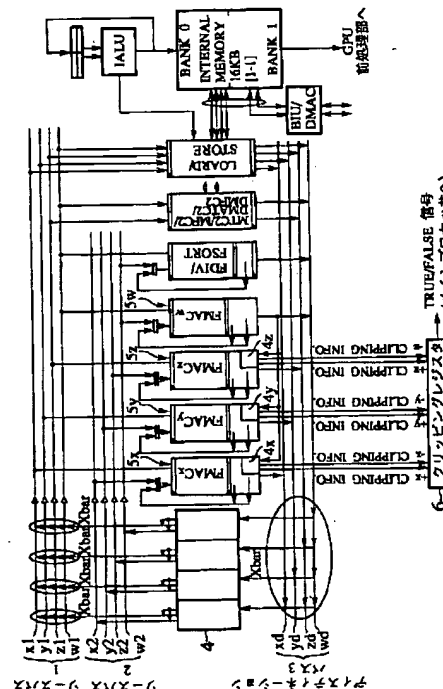
(74)代理人 弁理士 三好 秀和 (外3名)

(54)【発明の名称】 座標変換及びクリッピング判定を行う座標変換処理回路

(57) 【要約】

【課題】 この発明は、グラフィックス描画処理における座標変換後のクリッピング判定を効率良く行なう座標変換処理回路を提供することを課題とする。

【解決手段】 この発明は、座標変換後の出力座標ベクトル  $(x_n, y_n, z_n)$  と  $w_n$  のそれぞれの比較を 1 命令で行うクリッピング比較器をそれぞれの演算器 5 x, 5 y, 5 z に設け、クリッピング比較器の比較結果をクリッピングレジスタ 6 に集めてクリッピング判定を効率良く行なうように構成される。





## 【特許請求の範囲】

【請求項1】 入力座標ベクトル $(x, y, z, 1)$ の透視変換を行い、その変換結果として出力座標ベクトル $(x', y', z', w')$ の各要素を出力する4つの座標変換回路と、前記座標変換回路の夫々について設けられ、前記出力座標ベクトル $(x', y', z')$ の夫々の座標値を前記出力座標ベクトル $(w')$ の値と比較することによって、夫々の座標についてその表示領域内に収まっている否か判定する3つの比較回路と、前記比較回路からの出力を格納するクリッピングレジスタとを備えた座標変換処理回路。

【請求項2】 前記出力座標ベクトルの値は、夫々符号と絶対値を示すフィールドで表され、前記比較回路は、絶対値を示すフィールド同士を比較する比較器と、前記比較器からの出力と前記出力座標ベクトル $(x', y', z')$ の各々の座標値の論理積を出力する第1の回路と、前記比較器からの出力の反転信号と前記出力座標ベクトル $(x', y', z')$ の各々の座標値の論理積を出力する第2の回路とからなることを特徴とする請求項1に記載の座標変換処理回路。

【請求項3】 前記クリッピングレジスタはシフトレジスタであり、前記比較回路からの出力をシフト動作によって退避しておくこと特徴とする請求項1に記載の座標変換処理回路。

【請求項4】 前記比較回路は、 $x' > |w'|$ 又は $x' < -|w'|$ の場合に $x'$ が表示領域外であることを示す信号を出力し、 $y' > |w'|$ 又は $y' < -|w'|$ の場合に $y'$ が表示領域外であることを示す信号を出力し、 $z' > |w'|$ 又は $z' < -|w'|$ の場合に $z'$ が表示領域外であることを示す信号を出力すること特徴とする請求項1に記載の座標変換処理回路。

【請求項5】 前記比較回路の夫々は、前記出力座標ベクトル $(x', y', z')$ の夫々の座標値が、夫々の座標について表示領域内に収まっている場合と、その表示領域の正の方向に外れている場合と、その表示領域の\*

$$\begin{aligned} (x', y', z', w') &= (x, y, z, 1) \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} \\ &= (ax + ey + iz + lm, \\ &\quad bx + fy + jz + ln, \\ &\quad cx + gy + kz + lo, \\ &\quad dx + hy + lz + lp) \end{aligned}$$

となるので、4個の積和演算器を並列に4回繰り返せば座標変換を高速に実行することができる。この座標変換が終わった後、変換された座標が表示領域の範囲内にあるかどうかのクリッピングチェックを行なう。

【0004】 実際には座標変換後の1点 $(x', y', z', w')$ について考えると、

【数2】

\*負の方向に外れている場合とのいずれであることを示す2ビットの識別信号を出力することを特徴とする請求項4に記載の座標変換処理回路。

【請求項6】 前記クリッピングレジスタは、 $6 \times n$ ビット $(n > 1)$ のシフトレジスタであり、6ビット分のレジスタ部分が前記比較回路の識別信号を受け、6ビットのシフト動作によって順次他の部分に退避しておくこと特徴とする請求項5に記載の座標変換処理回路。

【請求項7】 前記クリッピングレジスタは、少なくとも18ビットのシフトレジスタであり、この18ビットの論理積を出力する出力回路が接続されていること特徴とする請求項6に記載の座標変換処理回路。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 この発明は、グラフィックス描画処理などに利用する座標変換を実行する浮動小数点プロセッサユニットの座標変換処理回路に関する。

【0002】

【従来の技術】 グラフィックス描画処理では、ポリゴンの頂点座標を3次元座標から表示装置用の2次元座標に変換する。この座標変換は $(4 \times 4)$ 行列と4要素のベクトルの積を計算することによって求められる。座標変換後の2次元座標は表示領域の範囲内に入っているか否かをチェックするクリッピング判定を行う必要がある。

【0003】 上記グラフィックス描画処理において、  
入力座標ベクトル :  $(x, y, z, 1)$

出力座標ベクトル :  $(x', y', z', w')$

座標変換・透視変換行列 :  $\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix}$

$\begin{pmatrix} e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix}$

$\begin{pmatrix} i & j & k & l \\ m & n & o & p \end{pmatrix}$

$\begin{pmatrix} m & n & o & p \end{pmatrix}$

とすると、

【数1】

$$-|w'| < x' < |w'| \quad \dots (1式)$$

$$-|w'| < y' < |w'| \quad \dots (2式)$$

$$-|w'| < z' < |w'| \quad \dots (3式)$$

上式全てが成立する場合に領域内となる。これを点のクリッピング判定と呼ぶことにする。

【0005】 領域外の場合には以下の条件でどの方向に出ているかチェックできる。

【0006】

\* \* 【数3】

 $|x'| < -|w'| : x \text{ 負方向の領域外} \quad \dots (4 \text{ 式})$ 
 $|x'| > |w'| : x \text{ 正方向の領域外} \quad \dots (5 \text{ 式})$ 
 $|y'| < -|w'| : y \text{ 負方向の領域外} \quad \dots (6 \text{ 式})$ 
 $|y'| > |w'| : y \text{ 正方向の領域外} \quad \dots (7 \text{ 式})$ 
 $|z'| < -|w'| : z \text{ 負方向の領域外} \quad \dots (8 \text{ 式})$ 
 $|z'| > |w'| : z \text{ 正方向の領域外} \quad \dots (9 \text{ 式})$ 

ポリゴン (n 角形) を描画する場合には n 頂点分の点クリッピング判定を行なう。その結果、全て領域内だった場合、そのポリゴンが領域内だと判定でき、そのまま表示する。一つでも頂点が領域外だと、そのポリゴンをクリップして表示する。また、そのポリゴンの全ての頂点が領域外の場合は二通りの場合がある。全ての頂点が一定方向の領域外なら、例えば全ての頂点が (4 式) を満たすならば、そのポリゴンを表示しない。それ以外、例えば頂点 1 が (4 式) を満たし、それ以外は (5 式) を満たす場合などは、そのポリゴンをクリップして表示する。

【0007】このような座標変換、クリッピング判定を行なう浮動小数点プロセッサの一例を図 5 に示す。

【0008】図 5 に示す浮動小数点プロセッサはコプロセッサとして設計されており、命令のフェッチ、デコードなどはメインプロセッサ側で行なわれる。条件分岐はコプロセッサ側から TRUE/FALSE 信号がメインプロセッサ側に送られ、その信号によってメインプロセッサ側で条件分岐が行なわれる。

【0009】この浮動小数点プロセッサは、128 ビットのソースバス 1 及びソースバス 2 と、同じく 128 ビットのディスティネーションバス 3 を有し、図 6 に示すようなレジスタ 4 ならびに図 7 に示すような演算器 7 x, 7 y, 7 z, 7 w を有している。それぞれの 128 ビットバスは 32 ビットづつ 4 グループに分けられる。これを x 系統、y 系統、z 系統、w 系統と呼ぶことにする。

【0010】ソースバス 2 とディスティネーションバス 3 では、x 系統のバスはレジスタ 4 のレジスタ番号が 4 で割り切れるレジスタ (R0, R4, R8, ...) に接続され、y 系統のバスはレジスタ番号が 4 で割ると 1 余るレジスタ (R1, R5, R9, ...) に接続され、z 系統のバスはレジスタ番号が 4 で割ると 2 余るレジスタ (R2, R6, R10, ...) に接続され、w 系統のバスはレジスタ番号が 4 で割ると 3 余るレジスタ (R3, R7, R11, ...) に接続されている。ソースバス 1 は全てのバスが全てのレジスタに接続されてクロスバー構成になっている。

【0011】演算器 7 x, 7 y, 7 z, 7 w はそれぞれ積和演算器 (FMAC x, FMAC y, FMAC z, FMAC w) を構成し、積和演算、乗算、加算、減算を 4 並列で実行することができる。比較演算などの特殊な演算は演算器 7 w が行なう。

【0012】図 5 に示すように接続することによって、座標変換における (4×4) 行列の計算を高速に行なうことができる。以下に、行列計算で用いる命令とプログラム例を示す。

【0013】記述：R 数値はレジスタ番号数値のレジスタを示す。

【0014】FMUL 命令

4 個の乗算を並列に行なう。

【0015】例：

FMUL R20-R23, R16, R0-R3

R20←R16×R0

R21←R16×R1

R22←R16×R2

R23←R16×R3

FMAC 命令

4 個の乗算を並列に行なう。

【0016】例：

FMUL R20-R23, R16, R0-R3

R20←R20+R16×R0

R21←R21+R16×R1

R22←R22+R16×R2

R23←R23+R16×R3

FNEG 命令

浮動小数点数の符号を反転する。

【0017】例：

FNEG R0, R1

R0←-(R1)

FCMP 命令

浮動小数点数の比較を行なう。比較結果は TRUE/FALSE 信号を用いてプロセッサ側に送られる。

【0018】例：

FCMP eq R0, R1

R0 と R1 が等しければ TRUE 信号を送る。そうでなければ FALSE 信号を送る。

【0019】FCMP ne R0, R1

R0 と R1 が等しくなれば TRUE 信号を送る。そうでなければ FALSE 信号を送る。

【0020】FCMP gt R0, R1

R0 が R1 より大きければ TRUE 信号を送る。そうでなければ FALSE 信号を送る。

【0021】FCMP le R0, R1

R0 が R1 以下ならば TRUE 信号を送る。そうでなければ FALSE 信号を送る。

【0022】FCMP lt R0, R1  
R0がR1より小さければTRUE信号を送る。そうでなければFALSE信号を送る。

【0023】FCMP eq R0, R1  
R0がR1以上ならばTRUE信号を送る。そうでなければFALSE信号を送る。

【0024】BCT命令  
コプロセッサからTRUE信号が送られて来たらLABELに分岐する。

【0025】例：  
BCT label

透視変換行列：

\*これらの命令を使って、三角形がポリゴンの座標変換とクリッピング判定を行なうプログラムを記述すると以下ようになる。

【0026】～各データとレジスタの対応～

変換前の三角形ポリゴンの頂点1：(R0, R1, R2, R3)

変換前の三角形ポリゴンの頂点2：(R4, R5, R6, R7)

変換前の三角形ポリゴンの頂点3：(R8, R9, R10, R11)

\*

R16, R17, R18, R19  
R20, R21, R22, R23  
R24, R25, R26, R27  
R28, R29, R30, R31

透視変換行列：

$$\begin{pmatrix} R16, R17, R18, R19 \\ R20, R21, R22, R23 \\ R24, R25, R26, R27 \\ R28, R29, R30, R31 \end{pmatrix}$$

変換後の三角形ポリゴンの頂点1：(R32, R33, 20※R38, R39)  
R34, R35)

変換後の三角形ポリゴンの頂点3：(R40, R41,

変換後の三角形ポリゴンの頂点2：(R36, R37, ※ R42, R43)

……ここからプログラムリスト……list 1……

；頂点1の座標変換

FMUL R32-R35, R0, R16-R19  
FMAC R32-R35, R1, R20-R23  
FMAC R32-R35, R2, R24-R27  
FMAC R32-R35, R3, R28-R31

；頂点2の座標変換

FMUL R36-R39, R4, R16-R19  
FMAC R36-R39, R5, R20-R23  
FMAC R36-R39, R6, R24-R27  
FMAC R36-R39, R7, R28-R31

；頂点3の座標変換

FMUL R40-R43, R8, R16-R19  
FMAC R40-R43, R9, R20-R23  
FMAC R40-R43, R10, R24-R27  
FMAC R40-R43, R11, R28-R31

；クリッピング判定(wは正と仮定)

；頂点1

FCMP gt R32, R35 ; 頂点1: x > wなら…  
BCT label ; クリッピング処理へジャンプ  
FCMP gt R33, R35 ; 頂点1: y > wなら…  
BCT label ; クリッピング処理へジャンプ  
FCMP gt R34, R35 ; 頂点1: z > wなら…  
BCT label ; クリッピング処理へジャンプ  
FNEG gt R35, R35 ; R35 = -(R35)  
FCMP lt R32, R35 ; 頂点1: x < -wなら…  
BCT label ; クリッピング処理へジャンプ  
FCMP lt R33, R35 ; 頂点1: y < -wなら…

```

BCT label          ;クリッピング処理ヘジャンプ
FCMP lt R34, R35    ;頂点1: z<-wなら…
BCT label          ;クリッピング処理ヘジャンプ
;頂点2
FCMP gt R36, R39    ;頂点2: x>wなら…
BCT label          ;クリッピング処理ヘジャンプ
FCMP gt R37, R39    ;頂点2: y>wなら…
BCT label          ;クリッピング処理ヘジャンプ
FCMP gt R38, R39    ;頂点2: z>wなら…
BCT label          ;クリッピング処理ヘジャンプ
FNEG R39, R39       ;R39=- (R39)
FCMP lt R36, R39    ;頂点2: x<-wなら…
BCT label          ;クリッピング処理ヘジャンプ
FCMP lt R37, R39    ;頂点2: y<-wなら…
BCT label          ;クリッピング処理ヘジャンプ
FCMP lt R38, R39    ;頂点2: z<-wなら…
BCT label          ;クリッピング処理ヘジャンプ
;頂点3
FCMP gt R40, R43    ;頂点3: x>wなら…
BCT label          ;クリッピング処理ヘジャンプ
FCMP gt R41, R43    ;頂点3: y>wなら…
BCT label          ;クリッピング処理ヘジャンプ
FCMP gt R42, R43    ;頂点3: z>wなら…
BCT label          ;クリッピング処理ヘジャンプ
FNEG R43, R43       ;R43=- (R43)
FCMP lt R40, R43    ;頂点3: x<-wなら…
BCT label          ;クリッピング処理ヘジャンプ
FCMP lt R41, R43    ;頂点3: y<-wなら…
BCT label          ;クリッピング処理ヘジャンプ
FCMP lt R42, R43    ;頂点3: z<-wなら…
BCT label          ;クリッピング処理ヘジャンプ
……ここまでプログラムリスト…… list 1……

```

## 【0027】

【発明が解決しようとする課題】以上説明したように、グラフィックス描画処理における座標変換を行う従来のプロセッサにおいては、二つの数値の比較命令と条件分岐命令を組み合わせるため、クリッピング判定のための比較と条件分岐に多くの命令を必要としていた。このため、クリッピング判定を伴う座標変換性能が低下してしまうという不具合を招いていた。

【0028】そこでこの発明は、上記に鑑みてなされたものであり、その目的とするところは、グラフィックス描画処理における座標変換後のクリッピング判定を効率良く行なう座標変換処理回路を提供することにある。

## 【0029】

【課題を解決するための手段】上記目的を達成するために、請求項1記載の発明は、入力座標ベクトル(x、y、z、1)の透視変換を行い、その変換結果としての出力座標ベクトル(x'、y'、z'、w')の各要素を出力する4つの座標変換回路と、前記座標変換回路の

夫々について設けられ、前記出力座標ベクトル(x'、y'、z')の夫々の座標値と前記出力座標ベクトル(w')の値を比較することによって、夫々の座標についてその表示領域内に収まっている否か判定する3つの比較回路と、前記比較回路からの出力を格納するクリッピングレジスタとを有して構成される。

【0030】請求項2記載の発明は、更に、前記出力座標ベクトルの値が、夫々符号と絶対値を示すフィールドで表され、前記比較回路は、絶対値を示すフィールド同士を比較する比較器と、前記比較器からの出力と前記出力座標ベクトル(x'、y'、z')の1つの座標値の論理積を出力する第1の回路と、前記比較器からの出力の反転信号と前記出力座標ベクトル(x'、y'、z')の1つの座標値の論理積を出力する第2の回路とから構成される。

【0031】請求項3記載の発明は、前記クリッピングレジスタはシフトレジスタであり、前記比較回路からの出力をシフト動作によって退避しておくことを特徴とす

る。

【0032】請求項4記載の発明は、前記比較回路は、 $x' > |w'|$  又は  $x' < -|w'|$  の場合に  $x'$  が表示領域外であることを示す信号を出力し、 $y' > |w'|$  又は  $y' < -|w'|$  の場合に  $y'$  が表示領域外であることを示す信号を出力し、 $z' > |w'|$  又は  $z' < -|w'|$  の場合に  $z'$  が表示領域外であることを示す信号を出力すること特徴とする。

【0033】請求項5記載の発明は、前記比較回路の夫々は、前記出力座標ベクトル ( $x'$ 、 $y'$ 、 $z'$ ) の夫々の座標値が、夫々の座標について表示領域内に収まっている場合と、その表示領域の正の方向に外れている場合と、その表示領域の負の方向に外れている場合とのいずれであるかを示す2ビットの識別信号を出力すること特徴とする。

【0034】請求項6記載の発明は、前記クリッピングレジスタは、 $6 \times n$  ビット ( $n > 1$ ) のシフトレジスタであり、6ビット分のレジスタ部分が前記比較回路の識別信号を受け、6ビットのシフト動作によって順次他の部分に退避しておくこと特徴とする。

【0035】請求項7記載の発明は、前記クリッピングレジスタは、少なくとも18ビットのシフトレジスタであり、この18ビットの論理積を出力する出力回路が接続されていること特徴とする。

【0036】

【発明の実施の形態】以下、図面を用いてこの発明の実施の形態を説明する。

【0037】図1に、本発明の1実施例による座標変換処理回路を用いたグラフィックス処理システムを示す。

【0038】このシステムでは、プロセッサ1と、グラフィックスプロセッサユニット（以下、「GPU」と記す。）3と、メインメモリ5がメインバス7でそれぞれ接続されており、さらに、グラフィックスメモリ9がGPU3にグラフィックスバス11で接続されている。

【0039】GPU3は前処理部13とメイン処理部15を有しており、メイン処理部15はDDAを使い、グラフィックスメモリ9から読み取ったテクスチャ情報を使って、出力画像のピクセルをレンダリングし、結果をグラフィックスメモリ9上のフレームバッファに書き込む。また、前処理部13はメイン処理のDDAの初期値、差分値などのパラメータを計算する。

【0040】プロセッサ1はプロセッサコア17によりメインメモリ5からプログラムを読み込んで実行する。プロセッサコア17はプログラムに従って、画面に描画すべきポリゴンの頂点の2次元座標と色情報等（以下、「GPUコマンド情報」と記す。）を作る。その際、座標変換ユニット19を使用して3次元座標から2次元座標を作る。さらに、プロセッサ1はプログラムに従って、上記GPUコマンド情報に付加情報（どんなポリゴンを描画するかを識別コマンド）を付け加え、GPUコ

マンドとしてメインバス7を通してGPU3に送り、GPU3が描画を実行する。なお、メインバス7を使わずに専用のバスを利用して情報を送る場合も考えられ得る。また、プロセッサ1のGPUコマンドを作るスピードと、GPU3の描画スピードがバランスしない場合（どちらかが遅い等）には、メインメモリ5上にGPUコマンドをバッファリングして速度差を吸収する。

【0041】図2に座標変換処理回路19の構成を示す。この実施形態の座標変換処理回路は、図5に示す座標変換処理回路に比べて、図5に示す演算器7x、7y、7zと同等の機能に加えて図3に示すクリッピング用比較器4x、4y、4zを内蔵した演算器5x、5y、5zと、図4に示すクリッピングレジスタ6を備えて構成されており、他の構成は図5と同様である。

【0042】図3において、クリッピング比較器4x、4y、4zは、座標変換後の頂点n ( $n = 1, 2, 3$ ) の浮動小数点による出力座標ベクトル ( $x_n, y_n, z_n, w_n$ ) を受けて、 $x_n$  と  $w_n$  を比較して  $x_n > |w_n|$  又は  $x_n < -|w_n|$  あるいは  $x_n \geq |w_n|$  又は  $x_n \leq -|w_n|$  を判別し、 $y_n$  と  $w_n$  を比較して  $y_n > |w_n|$  又は  $y_n < -|w_n|$  あるいは  $y_n \geq |w_n|$  又は  $y_n \leq -|w_n|$  を判別し、 $z_n$  と  $w_n$  を比較して  $z_n > |w_n|$  又は  $z_n < -|w_n|$  あるいは  $z_n \geq |w_n|$  又は  $z_n \leq -|w_n|$  を判別し、それぞれの判別結果を出力する動作を1つの比較命令により行い、座標変換後の頂点nの出力座標ベクトル ( $x_n, y_n, z_n, w_n$ ) を符号を含めて保持する入力レジスタ51、52と、入力レジスタ51に保持された出力座標ベクトル ( $x_n, y_n, z_n$ ) と入力レジスタ52に保持された出力座標ベクトル  $w_n$  の大小関係をそれぞれ比較する比較器53と、比較器53の比較結果と入力レジスタ51に保持された出力座標ベクトルの符号の反転値を入力して  $x_n > w_n$  又は  $y_n > w_n$  又は  $z_n > w_n$  あるいは  $x_n \geq w_n$  又は  $y_n \geq w_n$  又は  $z_n \geq w_n$  を判別する論理積 (AND) ゲート54と、比較器53の比較結果と入力レジスタ51に保持された出力座標ベクトルの符号を入力して  $x_n < -w_n$  又は  $y_n < -w_n$  又は  $z_n < -w_n$  あるいは  $x_n \leq -w_n$  又は  $y_n \leq -w_n$  又は  $z_n \leq -w_n$  を判別するANDゲート55を備えて構成されている。

【0043】図4において、クリッピングレジスタ6は、シフトレジスタ61と論理和 (OR) ゲート62を備えている。論理和 (OR) ゲート62は、シフトレジスタ61の右側の18ビットのビット積を出力する。

【0044】前記比較命令の実行により、シフトレジスタ61は、演算器5x、5y、5zのクリッピング比較器から出力されたそれぞれの判別結果を受ける為に、6ビット分左シフトを行う。つまり、一連の処理の中で、頂点 ( $n-1$ ) の前出力座標ベクトル ( $x_{n-1}, y_{n-1}, z_{n-1}, w_{n-1}$ ) の判別結果をシフトした後、次の

クリッピング判別結果を空いた右側の6ビットに受け取る。そして、判別結果が $n$ 頂点分（ここでは3頂点分）揃った段階で、シフトインされて保持された $n$ 角形（三角形）ポリゴンの頂点の判別結果からクリッピング比較器による判別が成立する頂点が存在するか否かのクリッピング判定が、シフトレジスタ61で行われ、クリッピングすべき頂点が存在する場合にはクリッピング処理を実行する分岐命令を実行させる信号を出力する。

【0045】図3に示すクリッピング比較器において、入力レジスタ51の入力は、演算器5xなら $x$ 値が、演算器5yなら $y$ 値が、演算器5zなら $z$ 値が入力される。入力レジスタ52の入力はどの演算器5x、5y、5zでも同一の $w$ 値が入力される。

【0046】入力 $x$ を代表して説明すると、入力 $x$ の符号ビットと、入力 $w$ の符号ビットを分離し、数値フィールドのみを比較器53に入力する。ここでは、入力 $x$ 、 $y$ 、 $z$ の数値フィールドを入力1とし、入力 $w$ の数値フィールドを入力2とする。ここで数値フィールドは、浮動小数点の指数部と仮数部からなっている。

【0047】比較器53の比較結果は、入力1>入力2の場合に「1」、そうでない場合には「0」となる。比較の条件は入力1 $\geq$ 入力2でも構わない。さらに、入力 $x$ 、 $y$ 、 $z$ の符号ビットの反転値をANDゲート54の一方の入力とし、入力 $x$ 、 $y$ 、 $z$ の符号ビットをANDゲート55の一方の入力とする。

【0048】図4に示す構成において、比較器53は入力 $x$ と入力 $w$ の絶対値を比較し、 $|x| > |w|$ で $x \geq 0$ （つまり、 $x > w$ ）の時、ANDゲート54の出力である+Clipping Info. が「1」になり、 $|x| > |w|$ で $x < 0$ （つまり、 $x < -w$ ）の時、ANDゲート55の出力である-Clipping Info. が「1」になる（ $w$ は正であると仮定する）。

【0049】次に、演算器5x、5y、5z、5wのそれぞれのクリッピング用比較器からの出力、+x Clipping Info.、+y Clipping Info.、+z Clipping Info.、-x Clipping Info.、-y Clipping Info.、-z Clipping Info.、をクリッピングレジスタ6への入力とする。

【0050】クリッピング用比較器は、FCLIP命令で演算器5x、5y、5zのクリッピング用比較器が同時に実行される。FCLIP命令は、6ビット左シフト後、クリッピング比較を行ない、クリッピング判定を行なう。詳しくは以下の通りである。

【0051】FCLIP命令動作例

FCLIP R0-R3, R4

(1) 第一動作

シフトレジスタ61を6ビット左シフト。

\* 【0052】(2) 第二動作

R0（入力 $x$ ）とR4（入力 $w$ ）を演算器5xのクリッピング比較器で比較する。R1（入力 $y$ ）とR4（入力 $w$ ）を演算器5yのクリッピング比較器で比較する。R2（入力 $z$ ）とR4（入力 $w$ ）を演算器5zのクリッピング比較器で比較する。

【0053】(3) 第三動作

クリッピング判定

シフトレジスタ61の下位18ビットに一つでも「1」があれば、メインプロセッサにTRUE信号を送る。

【0054】尚、上記第一動作と第二動作は、独立なので平行して行われる。

【0055】クリッピングレジスタ6を構成するシフトレジスタ61は、図5に示すように、例えば32ビットのレジスタで、最下位6ビットがクリッピング比較器の出力に接続されている。第0ビットが+Clipping Info. に、第1ビットが+Clipping Info. に、第2ビットが+Clipping Info. に、第3ビットが+Clipping Info. に、第4ビットが+Clipping Info. に、第5ビットが+Clipping Info. に、それぞれ接続されている。

【0056】シフトレジスタ61は、FCLIP命令が実行される度に6ビット左シフトされるので、常に最新の頂点のクリッピング情報が第0～5ビットに、一つ前の頂点のクリッピング情報が第6～11ビットに、二つ前の頂点のクリッピング情報が第12～17ビットに、三つ前の頂点のクリッピング情報が第18～23ビットに、四つ前の頂点のクリッピング情報が第24～29ビットに入っていることになる。

【0057】つまり、三角形ポリゴンの場合には3頂点分の情報で、ポリゴンのクリッピング判定を行なうので、ORゲート62を用いて第0ビット～17ビットの内一つでも「1」があるかどうかを判定し、「1」があればTRUE信号がメインプロセッサに出力され、クリッピング処理を行うべく分岐命令が実行される。

【0058】このような構成を採用することにより、クリッピング用比較器によって、二つ必要だったクリッピング判定用の比較命令を一つにでき、高速に処理できる。また、クリッピングレジスタ6によって、常に最新5頂点分のクリッピング情報を保持でき、三角形ポリゴンの3頂点分のクリッピング情報から、五角形ポリゴンの5頂点分のクリッピング情報を保持できる。さらに、上述したクリッピング用比較器とクリッピングレジスタを用いることにより、前述したlist 1は以下に示すlist 2のようになり、クリッピング判定のプログラムのステップ数を短縮することができる。

\* 【0059】

……………ここからプログラムリスト……………list 2……………

；頂点1の座標変換

```

FMUL  R32-R35, R0, R16-R19
FMAC  R32-R35, R1, R20-R23
FMAC  R32-R35, R2, R24-R27
FMAC  R32-R35, R3, R28-R31
      ; 頂点2の座標変換
FMUL  R36-R39, R4, R16-R19
FMAC  R36-R39, R5, R20-R23
FMAC  R36-R39, R6, R24-R27
FMAC  R36-R39, R7, R28-R31
      ; 頂点3の座標変換
FMUL  R40-R43, R8, R16-R19
FMAC  R40-R43, R9, R20-R23
FMAC  R40-R43, R10, R24-R27
FMAC  R40-R43, R11, R28-R31
      ; クリッピング判定 (wは正と仮定)
FCLIP R32-R34, R35; 頂点1のクリッピング情報をクリッピン
グレジスタへ
FCLIP R36-R38, R39; 頂点2のクリッピング情報をクリッピン
グレジスタへ
FCLIP R40-R43, R43; 頂点2のクリッピング情報をクリッピン
グレジスタへ
BCT   label          ; 最新3頂点分の情報を元にクリッピング
処理へジャンプ

```

.....ここまでプログラムリスト.....list 2.....

また、本実施例のクリッピングレジスタ6では、各頂点のクリッピング情報(+x, -x, +y, -y, +z, -z)を保持しているため、該当ポリゴンが全てクリッピング領域外にあるかどうかのチェックを効率良く行える。

【0060】例えば、三角形ポリゴンにおいては、頂点1の+xクリッピング情報ビット、頂点2の+xクリッピング情報ビット、頂点3の+xクリッピング情報ビットの全てが「1」の場合にそのポリゴンは全てクリッピング領域外にあり、表示する必要がないと判断できる。この場合、ORゲート62を頂点1の+x、頂点2の+x、頂点3の+xの論理積をとる論理ゲートに置き換えるようにすればよい。

【0061】尚、ここでは、座標変換後の頂点nを3個(n=1, 2, 3)としたが、任意の個数の頂点(n≥3)を持つ多角形のポリゴンも全く同様に取り扱えることは自明である。つまり、必要に応じて、ORゲート62の入力ビット数を増やし、又、シフトレジスタ61のビット幅を大きくすればよい。

【0062】

【発明の効果】以上説明したように、この発明によれば、座標変換後の出力座標ベクトル(xn, yn, zn)とwnのそれぞれの比較を1命令で行う構成を採用したので、クリッピング判定用の命令を少なくすることができ、かつクリッピング判定のプログラムのステップ数を削減できるので、クリッピング判定処理にかかる時

間を短縮することができる。また、クリッピング判定用の比較情報を専用のレジスタに集めてクリッピング判定を行う構成を採用したので、1ポリゴン分のクリッピング判定を効率良く行なうことができる。

【図面の簡単な説明】

【図1】本発明の1実施例による座標変換処理回路を用いたグラフィックス処理システムを示す。

【図2】発明の一実施形態に係る座標変換処理回路の構成を示す図である。

【図3】クリッピング比較器の構成を示す図である。

【図4】クリッピングレジスタの構成を示す図である。

【図5】グラフィックス処理における座標変換を行う従来の座標変換処理回路の構成を示す図である。

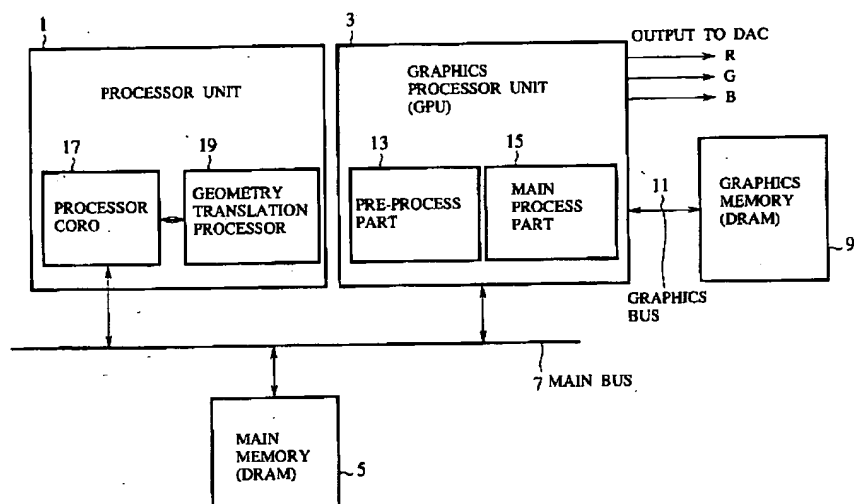
【図6】図5に示すレジスタの構成を示す図である。

【図7】図5に示す演算器の構成を示す図である。

【符号の説明】

1, 2 ソースバス  
 3 ディスティネーションバス  
 4 レジスタ  
 5 x, 5 y, 5 z, 5 w, 7 x, 7 y, 7 z, 7 w 演算器  
 6 クリッピングレジスタ  
 51, 52 入力レジスタ  
 53 比較器  
 54, 55, 62 論理ゲート  
 61 シフトレジスタ

【図1】



【図2】

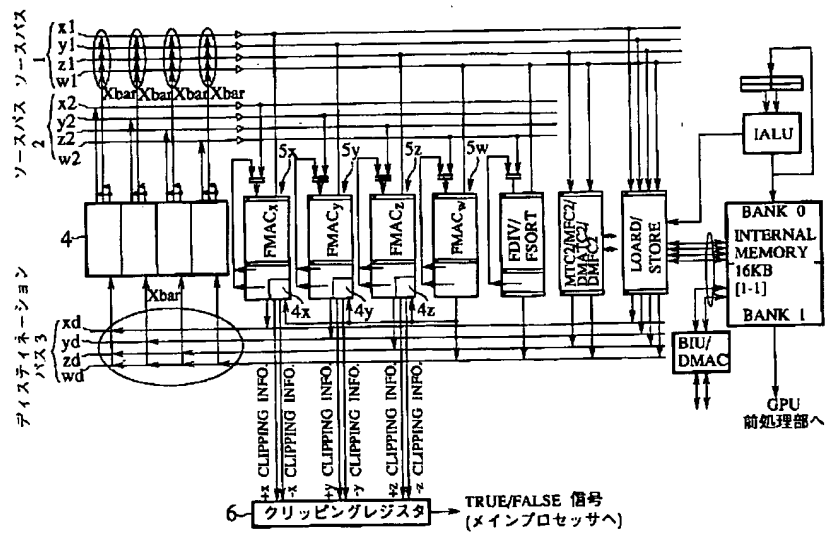


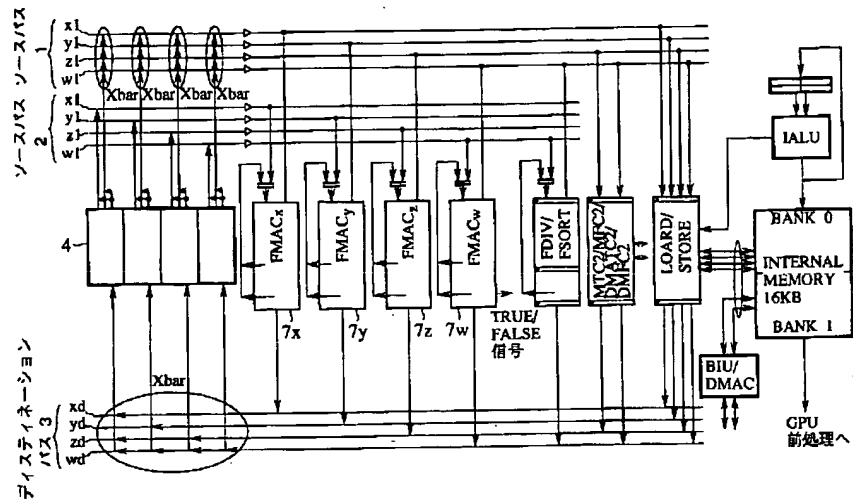


Figure 1 is a block diagram of a clipping information generation circuit. It consists of the following components and connections:

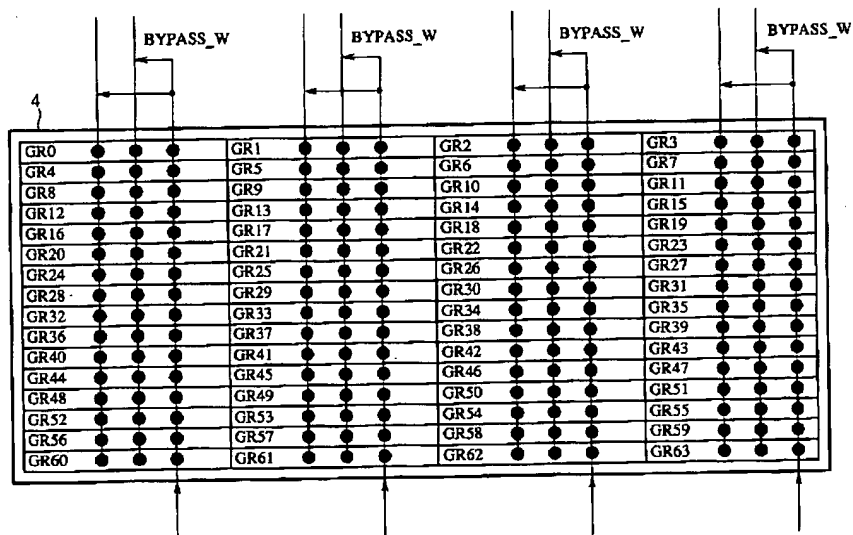
- Input 1 (Left):** Labeled "入力  $x(y,z)$ " (Input  $x(y,z)$ ). It includes a "符号ビット" (Sign Bit) and a "数値フィールド" (Numerical Field). The entire input is labeled "51".
- Input 2 (Right):** Labeled "入力  $w$ " (Input  $w$ ). It includes a "符号ビット" (Sign Bit) and a "数値フィールド" (Numerical Field). The entire input is labeled "52".
- Comparator (53):** Labeled "比較器" (Comparator). It receives "入力 1" (Input 1) and "入力 2" (Input 2) as inputs. It has two outputs: one leading down to the bottom of the page, and another leading to the left, which then branches to the inputs of AND gates 54 and 55.
- AND Gate 54:** Labeled "54". It takes the output from the comparator and the numerical field of input 1 as inputs. Its output is labeled "+CLIPPING INFO. ( $cx: +x, +y, +z$ )".
- AND Gate 55:** Labeled "55". It takes the output from the comparator and the numerical field of input 2 as inputs. Its output is labeled "-CLIPPING INFO. ( $cx: -x, -y, -z$ )".

[illegible]

【図5】



【図6】



【図7】

